**J505 Divya Dharshini Shankar**

**Banking System**

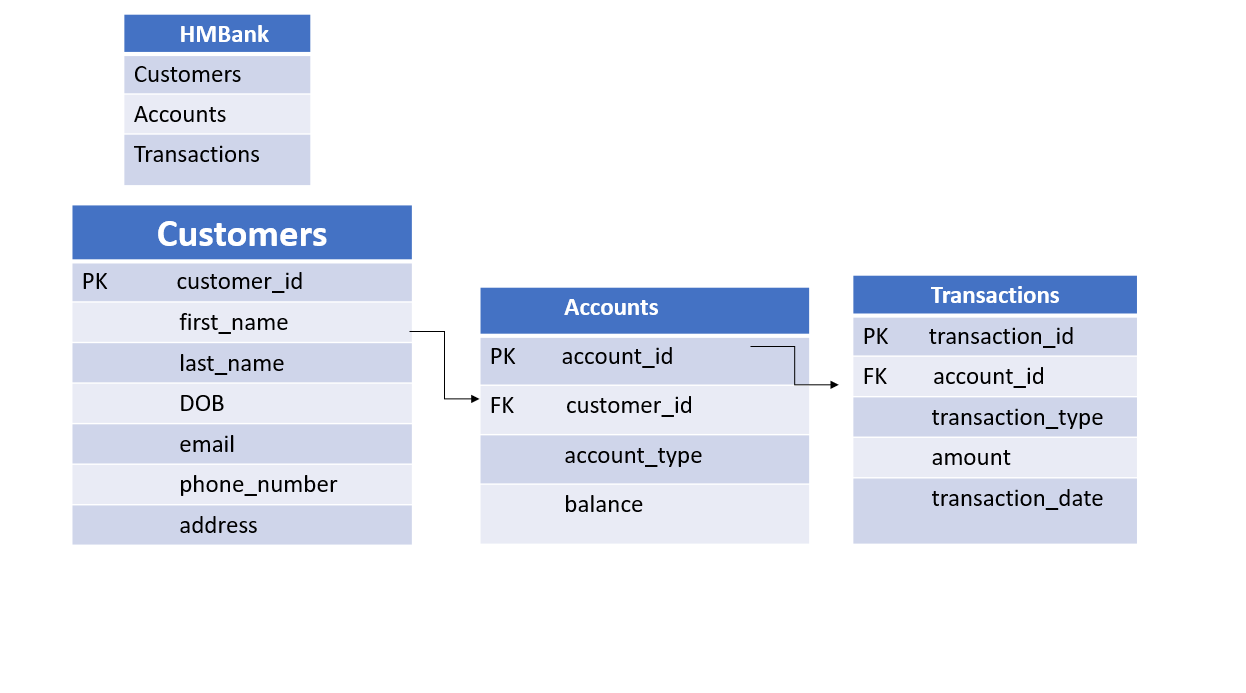
**1: Database Design:**

1.Create the database named "HMBank"

**Ans**: mysql>create database HMBank;

mysql>use HMBank;

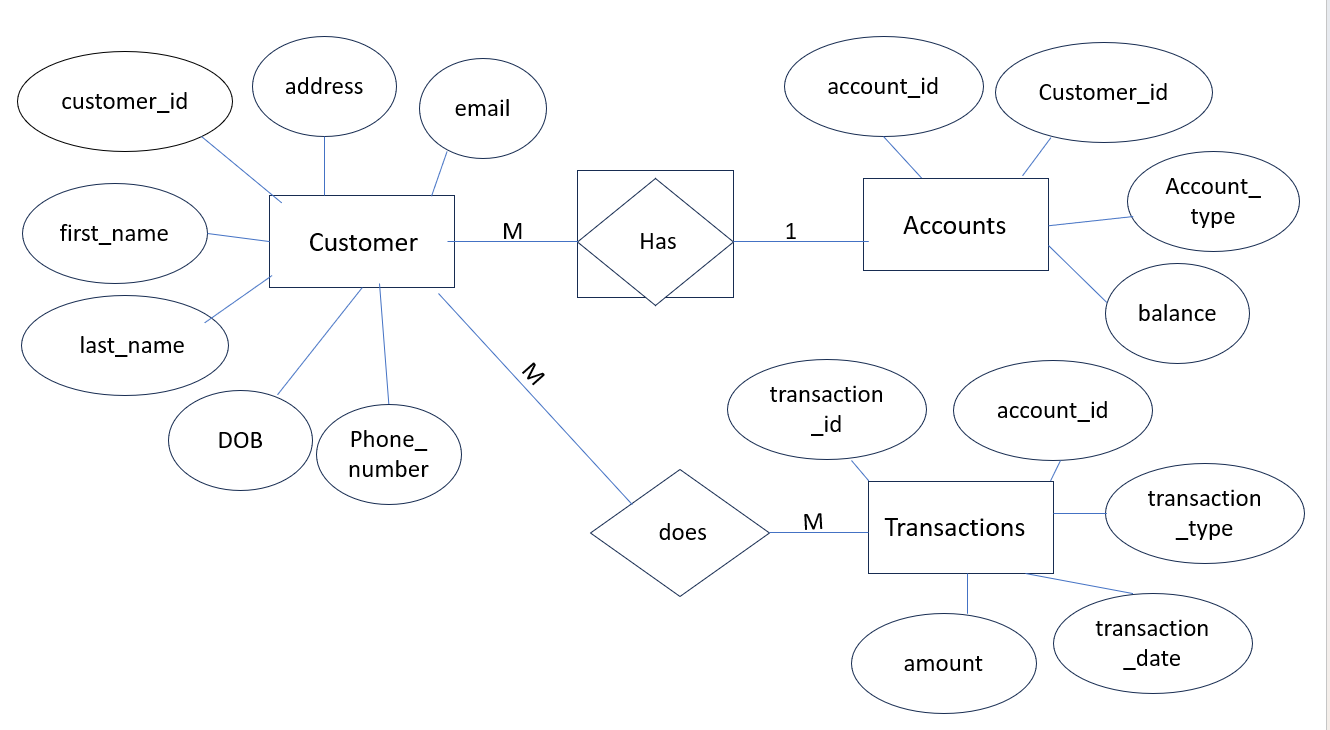
2. Define the schema for the Customers, Accounts, and Transactions tables based on the provided schema

**Ans:** mysql>create table Customers;

mysql>create table Accounts;

mysql>create table Transactions;

4.Create an ERD (Entity Relationship Diagram) for the database



6.Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships.

•Customers

**Ans:**

mysql>create table customers(customer\_id INT PRIMARY KEY,first\_name varchar

(30) NOT NULL, last\_name varchar (30) NOT NULL,DOB DATE,email varchar(30),phon

e\_number int, address varchar(50));

•Accounts

**Ans:**

mysql> CREATE TABLE Accounts (

-> account\_id INT PRIMARY KEY,

-> customer\_id INT,

-> account\_type VARCHAR(20),

-> balance INT,

-> FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id) );

•Transactions

**Ans:**

mysql> create table Transactions(transaction\_id INT PRIMARY KEY,

-> account\_id INT,

-> transaction\_type varchar(20),

-> amount INT,

-> transaction\_date DATE,

-> FOREIGN KEY (account\_id) REFERENCES Accounts(account\_id));

**Tasks 2: Select, Where, Between, AND, LIKE:**

1. Insert at least 10 sample records into each of the following tables.

•Customers

**Ans:**mysql> INSERT INTO Customers (customer\_id, first\_name, last\_name, DOB, email, phone\_number, address)

-> VALUES

-> (101, 'Abdul', 'Khan', '2000-06-13', 'abdul34@gmail.com', 6778899075, 'Chennai'),

-> (102, 'Badri', 'Kumar', '2001-04-03', 'badri6734@gmail.com', 6569169907, 'Lucknow'),

-> (103, 'Charu', 'Reddy', '1999-01-31', 'charu014@gmail.com', 9369169497, 'Andhra Pradesh'),

-> (104, 'Divya', 'Mahesh', '1998-09-17', 'divya187@gmail.com', 6569169907, 'Pune'),

-> (105, 'Eshwar', 'Pandey', '2001-08-10', 'eswari905@gmail.com', 6569169907, 'Mumbai'),

-> (106, 'Harini', 'Pandey', '2002-05-03', 'harini905@gmail.com', 6569169907, 'Salem'),

-> (107, 'Priya', 'Mithun', '2001-11-30', 'priya905@gmail.com', 6569169907, 'Bihar'),

-> (108, 'Vinay', 'Roy', '2002-12-05', 'vinay905@gmail.com', 6569169907, 'Pune'),

-> (109, 'Lokesh', 'Varma', '1995-08-29', 'lokesh905@gmail.com', 6569169907, 'Kolkata'),

-> (110, 'Komal', 'pandit', '2003-06-17', 'komal905@gmail.com', 6569169907, 'Chennai');

•Accounts

**Ans:**mysql> INSERT INTO Accounts (account\_id,customer\_id,account\_type,balance)

-> VALUES (786,101,"savings",8000),

-> (609,102,"current",70000),

-> (123,103,"savings",9000),

-> (564,104,"zero\_balance",0),

-> (409,105,"current",10000),

-> (904,106,"savings",3689),

-> (870,107,"current",7800),

-> (279,108,"savings",4560),

-> (322,109,"zero\_balance",0),

-> (455,110,"current",50000);

•Transactions

**Ans:**mysql> INSERT INTO Transactions (transaction\_id, account\_id, transaction\_type, amount, transaction\_date)

-> VALUES (489, 123, 'deposit', 6700, '2023-09-30'),

-> (490, 279, 'withdrawal', 4500, '2023-09-25'),

-> (491, 322, 'deposit', 12000, '2023-08-20'),

-> (492, 409, 'transfer', 8000, '2023-07-15'),

-> (493, 455, 'deposit', 5600, '2023-06-10'),

-> (494, 564, 'withdrawal', 3000, '2023-05-28'),

-> (495, 609, 'transfer', 7200, '2023-04-17'),

-> (496, 786, 'deposit', 15000, '2023-03-05'),

-> (497, 870, 'withdrawal', 2500, '2023-02-22'),

-> (498, 904, 'transfer', 9000, '2023-01-12');

**2. Write SQL queries for the following tasks:**

1. Write a SQL query to retrieve the name, account type and email of all customers.

**Ans:**mysql> select concat(c.first\_name,' ',c.last\_name) AS Name ,

-> c.email ,a.account\_type from customers c

-> join accounts a on c.customer\_id = a.customer\_id;

2. Write a SQL query to list all transaction corresponding customer.

**Ans:** mysql> select concat(c.first\_name,' ',c.last\_name) AS Name, t.transaction\_id,

-> t. account\_id,t.transaction\_type,t.amount,t.transaction\_date from

-> customers c join accounts a on c.customer\_id = a.customer\_id

-> join transactions t on a.account\_id = t.account\_id;

3. Write a SQL query to increase the balance of a specific account by a certain amount.

**Ans:** mysql> SELECT account\_id, balance, balance + 5000 AS "new balance"

FROM Accounts WHERE account\_id = 564;

4. Write a SQL query to Combine first and last names of customers as a full\_name.

**Ans:** mysql>select concat(first\_name,' ',last\_name) As full\_name from customers;

5. Write a SQL query to remove accounts with a balance of zero where the account type is savings.

**Ans:** mysql> delete from accounts where balance = 0 and account\_type = 'savings';

6. Write a SQL query to Find customers living in a specific city.

**Ans:** mysql> select concat(first\_name,' ',last\_name) AS Name from customers

-> where address = 'pune';

7. Write a SQL query to Get the account balance for a specific account

**Ans:** mysql> select balance from accounts where account\_id = 409;

8. Write a SQL query to List all current accounts with a balance greater than $1,000.

**Ans:** mysql> select \* from accounts where account\_type='current' and

-> balance >1000;

9. Write a SQL query to Retrieve all transactions for a specific account.

**Ans:** mysql> select \* from transactions where account\_id=609;

10. Write a SQL query to Calculate the interest accrued on savings accounts based on a given interest rate.

**Ans:**select account\_id, balance, balance \* 0.05 as interest\_accrued from accounts where account\_type = 'savings';

11. Write a SQL query to Identify accounts where the balance is less than a specified overdraft limit.

**Ans:** mysql> select \* from accounts where balance < 5000;

12. Write a SQL query to Find customers not living in a specific city.

**Ans:** mysql> select concat(first\_name,' ',last\_name) AS full\_name

-> from customers where address not in ('chennai');

**Tasks 3: Aggregate functions, Having, Order By, GroupBy and Joins:**

1. Write a SQL query to Find the average account balance for all customers.

**Ans**: mysql> select customer\_id,avg(balance) from accounts group by customer\_id;

2. Write a SQL query to Retrieve the top 10 highest account balances.

**Ans:** mysql>select balance from accounts

-> order by balance DESC;

3. Write a SQL query to Calculate Total Deposits for All Customers in specific date.

**Ans:** mysql> select sum(amount) from transactions where

-> transaction\_type='deposit' and transaction\_date ='2023-06-10';

4. Write a SQL query to Find the Oldest and Newest Customers

**Ans:** mysql> SELECT CONCAT(first\_name, ' ', last\_name) AS full\_name,DOB,

-> timestampdiff(year, dob, curdate()) as age

-> from customerswhere dob = (select min(dob) from customers)

-> or dob = (select max(dob) from customers);

5.Write a SQL query to Retrieve transaction details along with the account type.

**Ans:**mysql> select t.transaction\_id,t.account\_id, t.transaction\_type,

->t.amount,t.transaction\_date, a.account\_type

->from transactions t join accounts a on

->a.account\_id = t.account\_id;

6. Write a SQL query to Get a list of customers along with their account details.

**Ans:**mysql> select c.first\_name,c.last\_name,a.account\_id,

-> a.customer\_id,a.account\_type,a.balance from customers c

-> join accounts a on c.customer\_id = a.customer\_id;

7. Write a SQL query to Retrieve transaction details along with customer information for a specific account

**Ans:**mysql>select concat(c.first\_name,c.last\_name) AS full\_name,c.DOB,c.email,c.phone\_number, c.address,c.customer\_id, t.transaction\_id,t.account\_id, t.transaction\_type,t.amount,t.transaction\_date from customers c join accounts a on c.customer\_id = a.customer\_id join transactions t on a.account\_id = t.account\_id where a.account\_id=904;

8. Write a SQL query to Identify customers who have more than one account. ;

**Ans:**mysql> select c.first\_name,c.last\_name,c.customer\_id

from customers c join accounts a on c.customer\_id = a.customer\_id

group by c.customer\_id having count(a.account\_id) > 1 ;

9. Write a SQL query to Calculate the difference in transaction amounts between deposits and withdrawals.

**Ans:** mysql> select t.account\_id,t.amount as deposit\_amount,t1.amount as withdrawal\_amount,

(t.amount-t1.amount) as transaction\_amounts

from transactions t join transactions t1 on

t.account\_id=t1.account\_id

where t.transaction\_type ='deposit' and t1.transaction\_type='withdrawal';

10. Write a SQL query to Calculate the average daily balance for each account over a specified period.

Ans:mysql>select t.account\_id ,avg(a.balance) from accounts a join

transactions t on a.account\_id = t.account\_id where DATE(t.transaction\_date)=current\_date()

group by t.account\_id;

11. Calculate the total balance for each account type.

**Ans:**mysql> select sum(balance) as Total\_balance from accounts

-> group by account\_type;

12. Identify accounts with the highest number of transactions order by descending order.

**Ans:**mysql> select account\_id, count(\*) as total\_transactions from transactions

-> group by account\_id order by total\_transactions desc;

13. List customers with high aggregate account balances, along with their account types.

**Ans:**mysql> select concat(c.first\_name,' ',c.last\_name) as full\_name,

-> a.account\_type,a.balance from customers c join accounts a on

-> c.customer\_id = a.customer\_id

-> having balance>5000 order by a.balance desc;

14. Identify and list duplicate transactions based on transaction amount, date, and account.

**Ans:**mysql> select t.\* ,t1.\* from transactions t join transactions t1

-> on t.account\_id = t1.account\_id

-> where (t.amount!=t1.amount or

-> t.transaction\_date != t1.transaction\_date ) and

-> t.transaction\_id != t1.transaction\_id;

**Tasks 4: Subquery and its type:**

1. Retrieve the customer(s) with the highest account balance.

**Ans:** mysql>select c.\* from customers c join

->(select customer\_id, max(balance) as max\_balance from accounts

-> group by customer\_id) a on c.customer\_id = a.customer\_id

-> order by a.max\_balance desc limit 1;

2. Calculate the average account balance for customers who have more than one account.

**Ans:** mysql>select (a.balance) as avg\_balance from accounts a

-> join (select customer\_id from accounts group by customer\_id

-> having count(account\_id)>1) a1 on a.customer\_id != a1.customer\_id;

3. Retrieve accounts with transactions whose amounts exceed the average transaction amount.   
**Ans:**mysql> select account\_id ,amount from transactions where amount >

-> ( select avg(amount) as avg\_amount from transactions);

4. Identify customers who have no recorded transactions.

**Ans:** mysql>select customer\_id from customers where customer\_id not in

(select distinct customer\_id from accounts join transactions

on accounts.account\_id = transactions.account\_id );

5. Calculate the total balance of accounts with no recorded transactions.

**Ans:** mysql>select c.customer\_id ,(select sum(a.balance) from accounts a

where a.customer\_id = c.customer\_id) as total\_balance from customers c where c.customer\_id not in ( select distinct a.customer\_id from accounts a

join transactions t on a.account\_id = t.account\_id );

6. Retrieve transactions for accounts with the lowest balance.

**Ans:** mysql>select t.\*,a.account\_id,a.balance from transactions t join

(select account\_id,balance from accounts where balance =(select min(balance) from accounts ) ) a on a.account\_id = t.account\_id;

7. Identify customers who have accounts of multiple types.

**Ans**: mysql> select customer\_id from accounts where customer\_id in

-> (select customer\_id from accounts group by customer\_id

-> having count(Distinct account\_type)>1);

8. Calculate the percentage of each account type out of the total number of accounts.   
**Ans**: mysql> select account\_type, count(account\_id) as total\_accounts,

(count(account\_id) \* 100.0 / (select count(\*) from accounts)) as percentage

from accounts group by account\_type;

9. Retrieve all transactions for a customer with a given customer\_id.   
**Ans:** mysql> select a.customer\_id,a.account\_id,t.\* from transactions t join

(select customer\_id,account\_id from accounts where customer\_id =103)

a on a.account\_id=t.account\_id;

10. Calculate the total balance for each account type, including a subquery within the SELECT clause.

**Ans**:mysql>select distinct a.\*, a1.total\_balance from accounts a

join ( select account\_type, sum(balance) as total\_balance

from accounts group by account\_type )

a1 on a.account\_type = a1.account\_type;